

Recuperación de servicios Web basada en propiedades no funcionales

Web service retrieval based on non-functional properties

ESTEBAN GUERRERO

Ingeniero en Electrónica y Telecomunicaciones, magister en Ciencias de la Computación, estudiante de doctorado en Ciencias de la Computación. Investigador de la Universidad Umeå. Umeå, Suecia.

Contacto: gestevan@unicauca.edu.co

JUAN CARLOS CORRALES

Ingeniero en Electrónica y Telecomunicaciones, doctor en Ciencias de la Computación. Docente de la Universidad del Cauca. Popayán, Colombia.

Contacto: jcorral@unicauca.edu.co

Fecha de recepción: 31 de agosto de 2011

Fecha de aceptación: 17 de abril de 2012

Clasificación del artículo: Investigación

Palabras clave: emparejamiento de grafos, recuperación de servicios, servicios sensibles al contexto, servicios Web semánticos.

Key words: graph matchmaking, service retrieval, context-aware services, semantic-Web services.

RESUMEN

Actualmente, los sistemas de recuperación de servicios Web semánticos se basan en la utilización de las descripciones de las entradas, salidas, precondiciones y resultados, dejando de lado sus propiedades no funcionales. En este artículo se define una arquitectura experimental para la recu-

peración de servicios Web, la cual mejora el proceso de descubrimiento y recuperación de servicios Web semánticos basándose en la información que describe los requerimientos del usuario y las propiedades no funcionales del servicio. La arquitectura propuesta presenta un método tecnológicamente alternativo a los estándares actuales para describir y manejar los parámetros antes mencio-

nados. En esta arquitectura se utilizaron técnicas de emparejamiento basada en grafos, con el fin de facilitar la comparación y recuperación de servicios Web semánticos.

ABSTRACT

Nowadays, Semantic Web Services recovery systems are based on the use of descriptions of inputs, outputs, preconditions, and results; neglecting some existing non-functional properties. In this paper, we define an experimental architec-

ture (intended for Web service recovery) that improves the process of semantic-web-service discovery and retrieval based on the information that describes both user requirements and service-associated non-functional properties. The proposed architecture represents a technological alternative to current standards aimed at describing and managing non-functional properties. In this architecture, graph matchmaking techniques were used to compare and recover semantic Web services.

* * *

1. INTRODUCCIÓN

El ciclo de vida de los servicios Web comprende las siguientes actividades: creación, descripción, publicación, descubrimiento e invocación del servicio [1], algunas de estas requieren un intercambio de información explícito entre un humano y un sistema de recuperación, como es el caso del proceso de descubrimiento y selección de un servicio. El mejoramiento de estos dos procesos puede lograrse con la agregación de anotaciones semánticas y la inclusión de descripciones de características de calidad de servicio (Quality of Service - QoS) [2], o de otra índole, a las descripciones no semánticas de los servicios Web que son provistas por tecnologías como WSDL.

OWL-S y WSMO son tecnologías actuales usadas para describir las características funcionales, no funcionales y comportamentales de los servicios semánticos. Sin embargo, las especificaciones de las descripciones no funcionales en los estándares WSMO y OWL-S son aún ambiguas [3], [4].

Por otro lado, en diferentes áreas, los sistemas de recuperación utilizan datos del perfil y el con-

texto de un usuario para personalizar, filtrar y seleccionar información que es relacionada a las preferencias de éste; actualmente, esta información es importante debido al incremento en el número de servicios e información en línea.

En este trabajo se presenta una arquitectura alternativa para recuperar servicios utilizando las propiedades no funcionales (PNF) de los servicios Web semánticos y la información de un usuario, seleccionando las coincidencias más cercanas a los requerimientos del usuario, utilizando una aproximación independiente del lenguaje de descripción del servicio Web semántico y proporcionando una plataforma para el mejoramiento del proceso de selección de servicios Web. La arquitectura planteada hace uso de árboles (grafos) para la estructuración de los datos de las PNF y los requerimientos del usuario, utilizando, además, técnicas de emparejamiento de grafos para la comparación de tales estructuras.

La evaluación experimental de esta propuesta prueba que se pueden recuperar y clasificar servicios de manera óptima, utilizando parámetros de calidad de servicio requeridos por el usuario e información real de PNF de los servicios.

2. METODOLOGÍA

Para alcanzar el objetivo final de este proyecto, que es la recuperación de servicios web basado en propiedades no funcionales, se planteó un conjunto articulado de actividades para el desarrollo de los diferentes componentes de este proyecto. La metodología planteada está basada en los lineamientos señalados en el “Modelo Integral para un Profesional en Ingeniería” [18] de la siguiente manera: para el desarrollo de la base conceptual se utilizó el Modelo de Investigación Documental como herramienta para la síntesis de los referentes teóricos. El Modelo de Investigación Científica se utilizó como referente en la generación de las pautas y la definición de la arquitectura de mejoramiento del proceso de búsqueda de servicios utilizando mecanismos de personalización. El Modelo para Construcción de Soluciones fue empleado en el desarrollo de los prototipos de validación, haciendo uso de la metodología Rational Unified Process.

2.1 Metamodelos

La información de las PNF, de un servicio Web semántico, es extendida en éste trabajo utilizando metamodelos de información como se presentan en [12]. Esta información es utilizada para crear una descripción detallada de un servicio, con el fin de facilitar la correspondencia de los datos del usuario con la información de su descripción.

La utilización de este tipo de metamodelos permite describir diferentes categorías de información (QoS, contexto o perfil) y también definir para cada uno de sus parámetros un conjunto par *Valor - Métrica*, como lo presentan los modelos de la figura 1. Por lo tanto, cada servicio estará descrito por una o más categorías, ya sea información de QoS, perfil o contexto, cada una con diferentes dimensiones y parámetros. Las dimensiones del meta modelo de QoS definen categorías de conjuntos de información relacionadas con diferentes conjuntos de parámetros y métricas distintas [12].

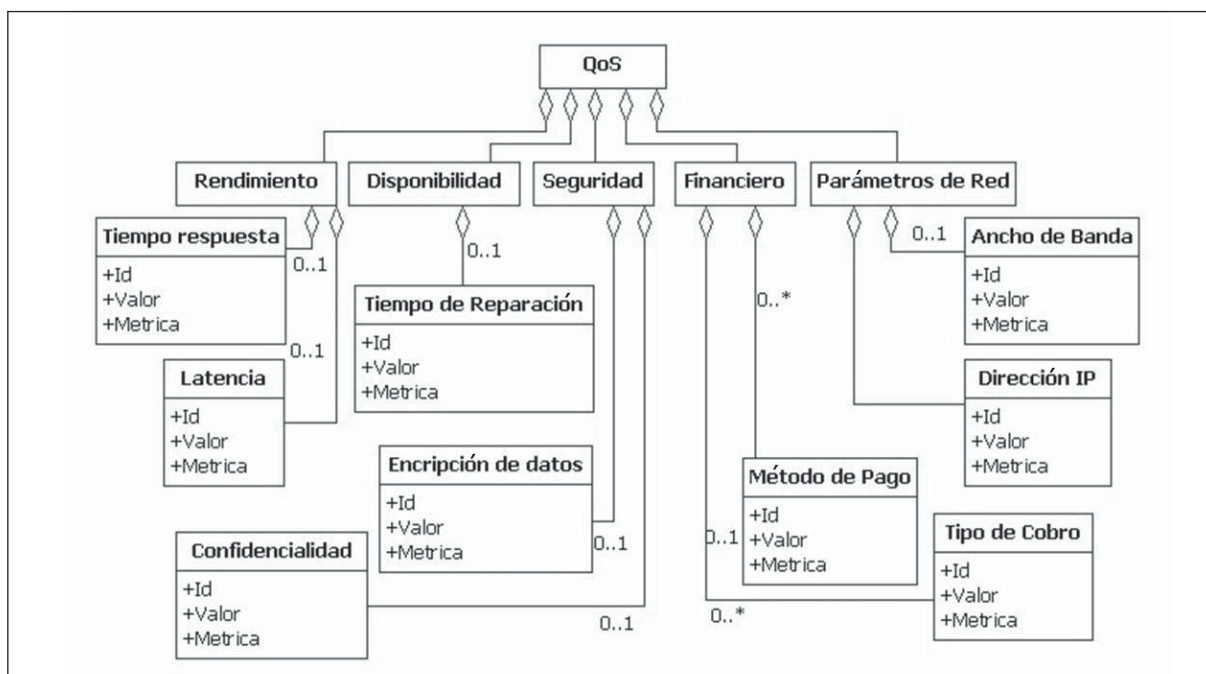


Figura 1. Meta-modelo de QoS [12].

Fuente: elaboración propia

Al respecto, se puede aclarar que esta propuesta estuvo centrada en el análisis de la información de QoS de un servicio para mejorar el proceso de selección, comparando los datos de los requerimientos de un usuario con la información de QoS de un servicio. Así, la definición de un meta modelo de QoS es un punto clave en el proceso de descripción de información detallada de PNF de un servicio Web semántico.

El metamodelo de la información de QoS de un servicio fue desarrollado teniendo en cuenta el trabajo previo [12] y su extensión para este trabajo fue basado en la recomendación de la W3C para servicios Web y la definición del estándar CC/PP (Composite Capability/Preference Profiles).

2.2 Representación semántica

El siguiente paso fue crear un modelo de datos para las PNF utilizando grafos. Dicho modelo es flexible y extensible; y permite describir la información de usuarios y servicios a la vez. El tipo de grafo utilizado es denominado árbol, cuya múltiple integración es llamada bosque [19], todos conectados a un nodo base como se presenta en la figura 2. La utilización de árboles en esta estructura de datos, representa entonces un conjunto de PNF (en este caso parámetros de QoS) de servicios conformando un bosque de PNF de servicios.

Un punto clave en el diseño de la estructura del árbol es la información del nombre del servicio, su descripción, los valores y métricas de cada parámetro del servicio; toda esta información es representada como pares valor-atributo, localizados en los nodos del árbol. La URI de acceso al servicio es un dato clave en este caso ya que ésta identifica al servicio Web al cual pertenecen las PNF. Esta información es utilizada como un índice de referencia de acceso a la estructura de datos, almacenándola en cada relación (arista) del

mismo grafo. Por consiguiente, esta propuesta define un árbol como un servicio y todas las relaciones dentro del grafo tienen como parámetro la URI, como es representado en la figura 2. En esta figura se presenta la estructura del árbol que contiene la información de los grafos, cada nivel contiene nodos con diferente información a partir de la raíz. En el nivel de *servicios* se encuentra la descripción de un servicio como de un usuario con un valor de identificación. En el nivel de categorías se definen tres tipos: QoS, perfil y contexto, siguiendo el modelo de [12]. Las dimensiones reflejan subconjuntos definidos de datos/atributos relacionados, que para este trabajo de investigación fueron la dimensión de rendimiento y disponibilidad. La información de *VALOR* y *MÉTRICA* de cada parámetro, de su dimensión y de las categorías QoS, contexto o perfil es almacenada en las hojas de cada árbol como se presenta en la figura 2. La información del nombre y descripción de cada servicio es almacenada en el parámetro *nombre del servicio* conectado directamente al nodo base.

Al igual que los metamodelos planteados en [12], los servicios pueden tener o no diferentes categorías (QoS, Contexto o Perfil) al mismo tiempo y por lo tanto un mismo árbol puede tener una sola dimensión de una categoría o múltiples categorías con múltiples dimensiones, describiendo diferentes tipos de información en un mismo modelo de datos. Este tipo de modelo de datos, definido como un grafo, permite extender la descripción de un servicio de manera flexible, definiendo valores y métricas de múltiples parámetros al mismo tiempo sin afectar o interactuar con otros parámetros de otras dimensiones.

El modelo de datos en árbol escogido para representar la información de los datos tiene una estrecha relación con los metamodelos planteados en [12], los metamodelos son representados como diferentes categorías (QoS, Contexto o Perfil) como se representa en la figura 3. A nivel de implemen-

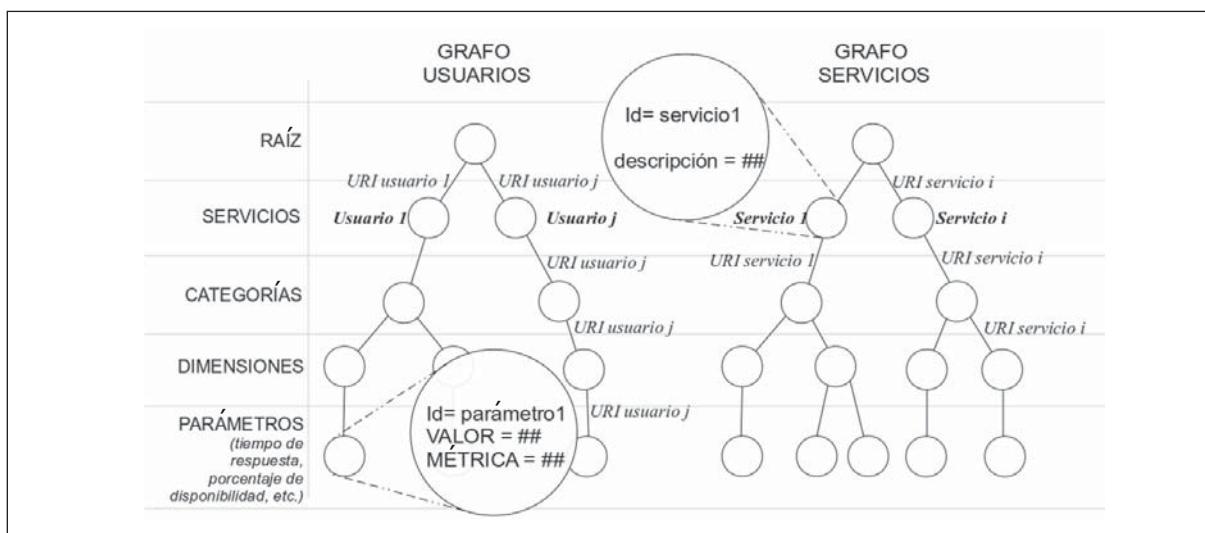


Figura 2. Modelo de datos de los grafos de los requerimientos del usuario y de las PNF del servicio.

Fuente: elaboración propia

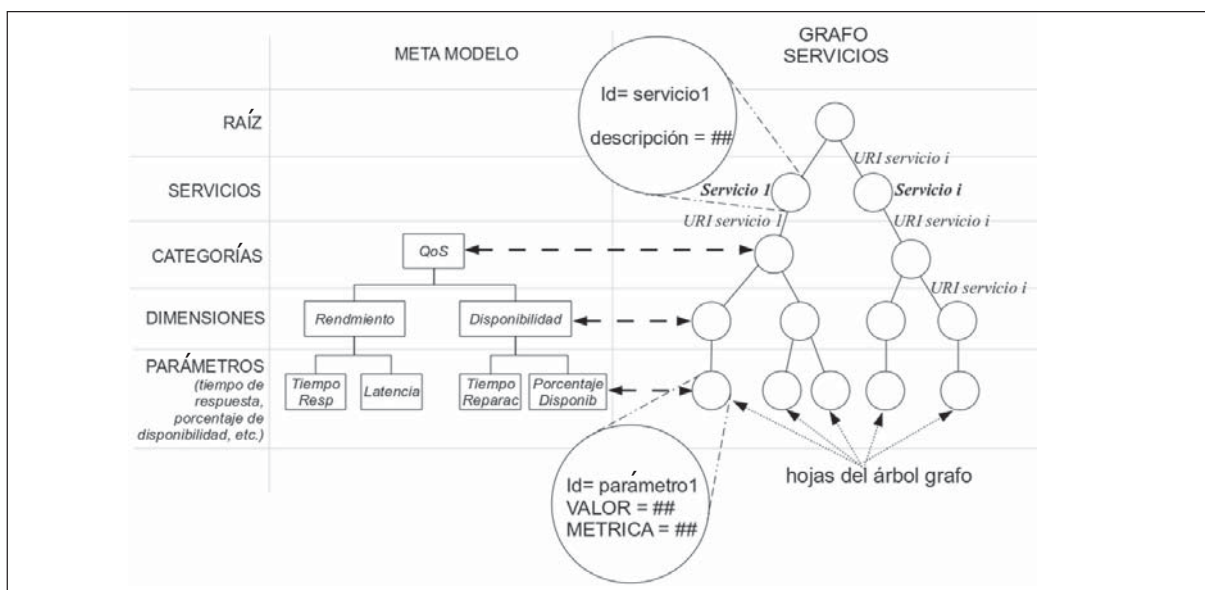


Figura 3. Relación existente entre el meta modelo y el modelo de datos en forma de árbol.

Fuente: elaboración propia

tación, la información extra, representada en los grafos, puede ser almacenada y condensada utilizando RDF y RDF *schema* ya que éstos representan de forma natural modelos de grafos, permitiendo almacenar información de las PNF de un

servicio o de múltiples servicios como un repositorio de PNF en archivos RDF – RDF *schema*.

Al transformar la descripción de las PNF de un servicio Web a un modelo de grafos, se obtiene

un modelo independiente del lenguaje, de la descripción utilizada en la definición del servicio y, al mismo se tiempo, consigue una compatibilidad con los estándares actuales.

2.3 Emparejamiento de propiedades no funcionales

Para realizar la comparación de la información almacenada en los dos conjuntos de grafos que representan las PNF de los servicios y la información de los requerimientos no funcionales de los usuarios, se creó un algoritmo de emparejamiento de grafos con base en el algoritmo presentado en [20]. En este nuevo algoritmo (tabla 1) se analizó la estructura de cada árbol, utilizando para tal fin algoritmos de comparación isométrica de grafos.

El algoritmo propuesto utiliza la misma estrategia que el algoritmo VF2 [20]. VF2 crea un nuevo grafo que contiene los nodos que coinciden con los parámetros requeridos del grafo del usuario U_j . En este caso, dicho grafo incluye las PNF S_i , utilizando estados (s) para analizar paso a paso el proceso de emparejamiento. El algoritmo de búsqueda que recorre todos los nodos del árbol es el algoritmo *depth-first*; en la aproximación propuesta, este algoritmo es aplicado a través de los nodos del grafo del servicio seleccionado, utilizando la URI como información de indexado; en éste proceso de indexado se obtienen, parametrizan y se almacenan datos que facilitan una rápida y precisa recuperación de información. Este tipo de técnica es común a muchas de los sistemas basados en grafos [21]. El uso de *depth-first* se debe a que cada nodo del árbol conectado contiene un nodo raíz [22] y la altura de cada árbol del servicio o del usuario es corto (5 niveles siguiendo el meta modelo de [12]), razón por la cual la búsqueda en profundidad lleva poco tiempo en ejecutarse.

La aplicación del algoritmo de la tabla 1 a un par grafo de un usuario, grafo de un servicio, trae

como resultado la creación de otro grafo el cual se ha denominado grafo M (M por su correspondencia en inglés a Matching), este grafo contiene los nodos que el grafo del servicio tiene en común con el grafo del usuario. Esto se genera a través de la utilización de varias funciones dentro del algoritmo (ver tabla 1) como *comparahojas_METRICA*, que es una función que permite establecer si los parámetros METRICA dentro de las hojas del árbol de U_j y las hojas de S_i son iguales, en cuyo caso se analiza el parámetro VALOR sino se aplica la función de normalización de dichas

Tabla 1. Algoritmo de isomorfismo

1.	Para cada: Si
2.	Depth_first(U_j)
3.	Depth_first(S_i)
4.	Para cada: P(s)
5.	compara(nombre_deHoja(U_j y S_i)):
6.	Si: nombre_deHoja U_j = nombre_deHoja S_i
7.	comparahojas_METRICA(U_j y S_i):
8.	Si: hoja_METRICA U_j = hoja_METRICA S_i
9.	computa hoja_PESO M(s)
10.	agrega nodo hoja M(s) a M
11.	Si: hoja_METRICA U_j != hoja_METRICA S_i
12.	normaliza_METRICA(U_j , S_i)
13.	computahoja_PESO M(s)
14.	agrega nodo hoja M(s) a M
15.	Fin de bucle Para
16.	Fin de bucle Para
<p>Si= árboles del servicio: bosque</p> <p>U_j= árbol de la información del usuario</p> <p>M= grafo de emparejamiento, es el grafo resultante de la comparación de U_j y S_i</p> <p>M(s) = grafo de emparejamiento parcial en el estado s</p> <p>Depth_first: algoritmo depth first para cada S_i utilizando la URI as camino.</p> <p>comparahojas_METRICA: es una función para analizar la respectiva métrica de cada hoja del árbol.</p> <p>computahoja_PESO: función que define el valor PESO del grafo M, esto es la proximidad entre U_j y S_i</p> <p>normaliza_METRICA: normaliza los valores de METRICA</p>	

Fuente: elaboración propia

métricas; *normaliza_METRICA* es una función que normaliza los parámetros METRICA de cada hoja, en la arquitectura prototipo desarrollada dicha función estandariza métricas de Tiempo de Respuesta del servicio (transformaciones entre diferentes tasas de velocidad bytes/segundo, kilo bytes/segundo y megabytes/segundo) esto es un punto clave para la aproximación propuesta, y su implementación depende de la naturaleza de los parámetros que se utilicen en los meta modelos.

Otra de las funciones utilizadas (tabla 1) fue *computa_hoja_PESO*, que es una función para determinar la similitud que tiene cada hoja del grafo de servicio con su respectiva correspondencia del grafo del usuario, esta función toma el parámetro VALOR de U_j y S_p , y utiliza algunas reglas detalladas en la sección siguiente. La función fija el valor PESO, que es un nuevo parámetro que se agrega al grafo M_{ji} . Este parámetro permite seleccionar el grafo del servicio con mayor similitud al grafo del usuario y por consiguiente determinar el servicio más cercano a los requerimientos de un usuario.

En resumen, luego de aplicar el algoritmo de emparejamiento, el grafo M contendrá solo los grafos de los servicios que tengan parámetros similares a los del grafo del usuario; cada hoja del grafo M detalla información de similitud por medio del parámetro PESO; también declara información de la URL del servicio (contenida dentro de cada arista del grafo) de manera similar al grafo de servicio (ver figura 4). Finalmente, la estructura del grafo M utiliza el mismo diseño del grafo de servicios tipo bosque.

2.4 Proceso de filtrado y clasificación de servicios

En esta sección se detalla el proceso de computación del parámetro PESO y de los procedimientos para la selección de los grafos de servicio con mayor grado de similitud a los grafos de los requerimientos del usuario. El parámetro PESO se establece como una distancia relativa entre los parámetros de la hoja del servicio y la hoja del

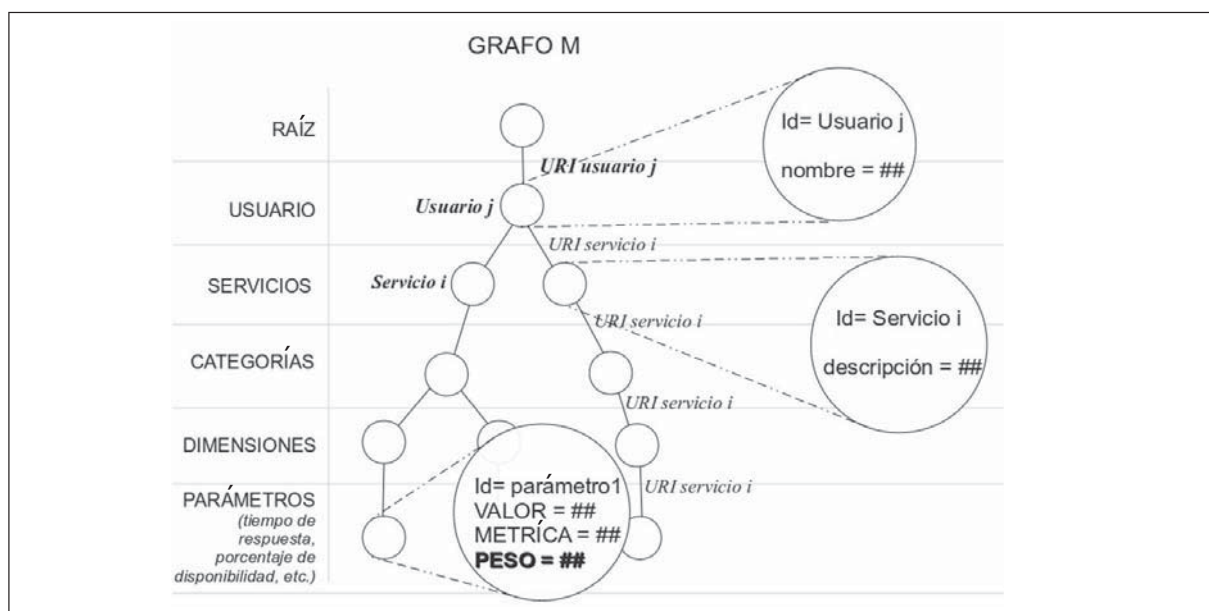


Figura 4. Relación existente entre el meta modelo y el modelo de datos en forma de árbol.

Fuente: elaboración propia

usuario; el proceso de selección de un grafo de servicio utiliza la misma abstracción de distancia relativa, utilizando los parámetros peso de todas las hojas del grafo.

Para comenzar el análisis se definen los conjuntos de nodos de los grafos, sea $S=\{S_1, S_2, \dots, S_i\}$ un conjunto de i servicios que contienen todos los árboles grafo con la información de las PNF de servicios semánticos, como se puede ver en la figura 2.

Sea también U_j un árbol grafo con requerimientos no funcionales de un usuario, definido de igual manera como S , con $U=\{U_1, U_2, \dots, U_j\}$. Así como M , un árbol grafo creado utilizando el algoritmo de la tabla 1, que contiene los servicios relacionados con U_j como subgrafos de éste, siendo este grafo definido por k y k el número de árboles relacionados con U_j . Cada hoja de contiene la información de VALOR, METRICA y PESO de cada parámetro l de las dimensiones del servicio.

Utilizando la información del parámetro PESO de cada hoja de M , se puede abstraer este parámetro como una distancia Euclidiana entre el dato de VALOR de un parámetro del nodo hoja de U_j y el dato de VALOR del parámetro del correspondiente nodo hoja de S_i , esta distancia es corregida utilizando la información de METRICA si es necesario.

El valor de PESO es definido por:

$$\omega_{l_{sk}} = |v_{l_{Uj}} - v_{l_{Si}}| \quad (1)$$

Donde $\omega_{l_{sk}}$ en la ecuación (1) es el VALOR de un parámetro l en el grafo U_j , y $v_{l_{Si}}$ es el valor de un parámetro l en el árbol grafo S_i . Esta propuesta hace uso de la distancia euclidiana debido a que en [12] los parámetros de los metamodelos son definidos como univariados e independientes entre sí, pero con el objeto de generalizar la solución, se sugiere utilizar un método diferente

de normalización y métrica, ya que los parámetros no tienen una naturaleza lineal, es decir, los parámetros de los grafos tienen dependencia de valores entre sí.

Con el objetivo de seleccionar los servicios que tengan mayor similitud con los parámetros de U_j , se han establecido un conjunto de cuantificadores que definen la relevancia de los parámetros contenidos en las dimensiones (definidos por el usuario o por el contexto) y que están relacionados con valores de *distancia de tolerancia* de los datos de VALOR de U_j , de esta manera se define el conjunto de valores de $v_{l_{Si}}$ que pueden ser admitidos por un usuario determinado.

Sea:

$$I = \{I_1, I_2, \dots, I_l\} \quad (2)$$

El vector de Importancia para cada parámetro l como se presenta en la ecuación (2) definido en los metamodelos de [12], con $I_l = [0,1]$. Con el objetivo de establecer una métrica para el conjunto de servicios relacionados, en la expresión se define un *valor de tolerancia* de $v_{l_{Uj}}$ denominado Δv_l , el cual es calculado utilizando el dato VALOR del parámetro l : $v_{l_{Uj}}$ y el valor respectivo de I_l como:

$$\Delta v_l = I_l v_{l_{Uj}} \quad (3)$$

Este valor de tolerancia es pensado debido a que los parámetros de selección de los servicios pueden ser no exactos, por ejemplo, cuando un usuario tiene un rango de preferencias por un parámetro o éste debe ser mayor o menor que un valor definido.

Con y es posible calcular el conjunto de (servicios relacionados al usuario); este conjunto es calculado por:

$$\omega_{l_{sk}} \leq \Delta v_{l_{Uj}} - v_{l_{Si}} \mid \Delta v_l = I_l v_{l_{Uj}} \quad (4)$$

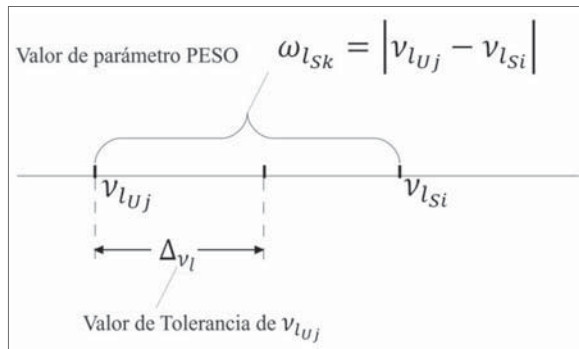


Figura 5. Distancia de tolerancia y el parámetro PESO.
Fuente: elaboración propia

El análisis de la ecuación (4) es representado en la figura 5.

Los desarrollos matemáticos previos presentados en la ecuación (1), la ecuación (2), la ecuación (3) y la (4) son definidos cuando el servicio contiene un solo parámetro l (una sola hoja), cuando más parámetros existen es posible abstraer el análisis de varios parámetros como dimensiones en un espacio l -dimensional, el cual define un espacio euclidiano de l dimensiones de la siguiente manera: sea X y Y vectores; luego, cualquier número real $\|X\|$ tal que:

- I. $\|X\| \geq 0, \|X\| = 0 \Leftrightarrow X = 0$;
- II. $\|\alpha X\| = |\alpha| \|X\|, \alpha \in \mathbb{R}$;
- III. $\|X + Y\| \leq \|X\| + \|Y\|$;

Se define como una distancia euclidiana entre los puntos X y Y , o la longitud de los vectores desde X hasta Y como [23]:

$$\|X + Y\| = \sqrt{[(x_1 - y_1)^2 + \dots + (x_n - y_n)^2]} \quad (5)$$

En este caso, el punto X es $(0, 0, \dots, 0)$ y Y es la distancia $\omega_{l_{Sk}}$, lo cual en un perfecto emparejamiento de los grafos U_j y S_i el valor de $\omega_{l_{Sk}}$ sería $\omega_{l_{Sk}} = 0$.

De la ecuación (5) se puede definir la distancia de un servicio S_i como:

$$W_{S_i} = \sqrt{[(\omega_1)^2 + (\omega_2)^2 + \dots + (\omega_l)^2]} \quad (6)$$

Expresando la ecuación (6) en una forma más general se tendría:

$$W_{S_i} = \sqrt{[\sum_{k=1}^l (\omega_k)^2]} \quad (7)$$

Y con el servicio seleccionado cumpliendo con la condición establecida por la ecuación (3). Ahora bien, se pueden catalogar los servicios utilizando la medida de la distancia euclidiana como se define en la ecuación (7), en l dimensiones (l es el número de parámetros de cada grafo servicio), siendo los servicios más cercanos a la información del usuario aquellos que tengan un valor cercano a cero.

3. RESULTADOS

3.1 Arquitectura de recuperación de servicios

La arquitectura para la recuperación de servicios propuesta consta de tres módulos principales: el módulo de gestión de la información del usuario, el módulo de gestión de información de los servicios Web semánticos, y el módulo de categorización y visualización de la información de personalización, como se presenta en la figura 6.

El *módulo de gestión de la información del usuario* obtiene la información del usuario en forma de requerimientos no funcionales que debe cumplir el servicio Web.

El *módulo de gestión de información de los servicios Web semánticos* se encarga de gestionar la información de las propiedades no funcionales del servicio Web; éste extrae u obtiene información de las PNF y crea archivos descriptores (RDFs) externos que contienen la información de las PNF

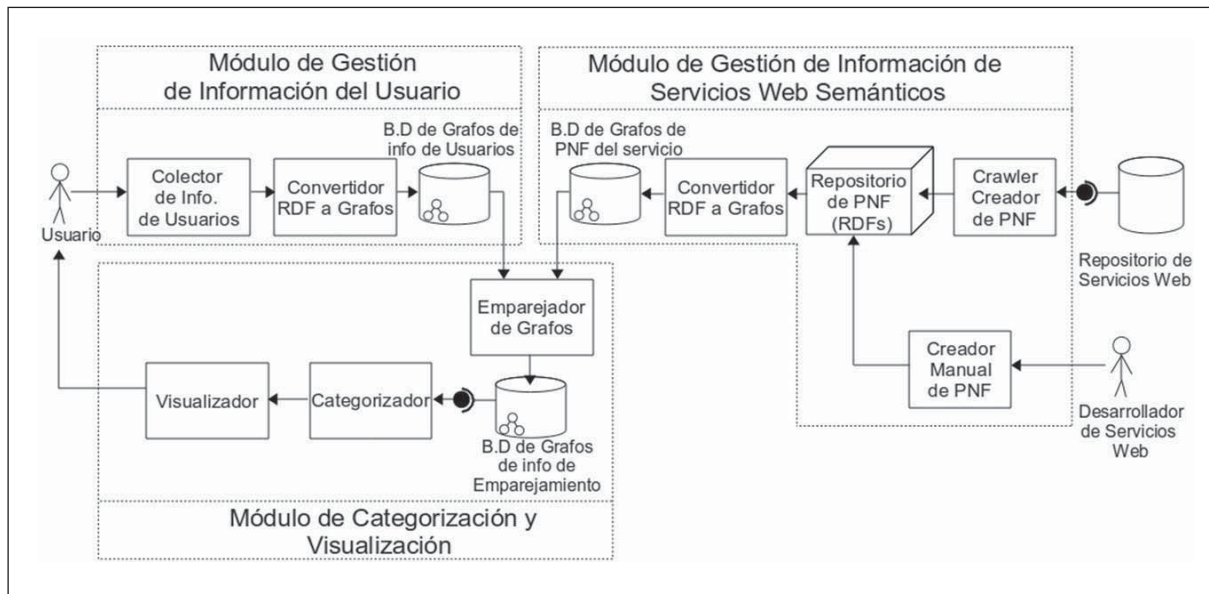


Figura 6. Arquitectura detallada de recuperación de servicios.

Fuente: elaboración propia

siguiendo los meta modelos de [12], enlazando los descriptores originales (WSMO o OWL-S) con los archivos RDF.

Una vez convertidas las representaciones a grafos, tanto de la información de requerimientos de los usuarios como de las PNF de los servicios (árboles de información), el *módulo de categorización y visualización de la información de personalización* se encarga de categorizar los servicios que tengan mayor similitud con la información del usuario, utilizando para tal efecto algoritmos de emparejamiento de grafos y algoritmos específicos de categorización (*ranking*) de información.

Con el objeto de obtener información real de QoS de un servicio, fueron diseñados dos submódulos *Creadores de PNF* (bloques del módulo de Gestión de Información de SWS de la figura 6), los cuales obtienen información de las PNF de un servicio, desde un repositorio por medio de un proceso de *rastreo* (*crawling*) o a través del ingreso manual de estos datos por parte del *desarrollador de servicios Web*

El submódulo *Crawler* obtiene información de las PNF utilizando un software que extrae información a partir de un conjunto de URLs, detectando las URIs de los servicios y extrayendo información de QoS como tiempo de respuesta y porcentaje de disponibilidad del servicio desde los servidores que los albergan. El submódulo *Manual* contiene un formulario con el cual un diseñador o desarrollador del servicio puede manualmente ingresar datos de QoS.

Después de obtenida la información de las PNF, el submódulo *Repositorio de PNF (RDFs)* crea los archivos RDF de descripción que luego son transformados a árboles y almacenados en la base de datos de grafos de las PNF del servicio (*B.D de Grafos de PNF del servicio* de la figura 6).

Por otro lado, en el módulo de gestión de la información del usuario, el submódulo de obtención de información del usuario (*Colector de Info. de Usuarios*), captura los requerimientos de QoS por parte del usuario, que son transformados a grafos (árboles) utilizando el mismo proceso descrito

anteriormente de las PNF en los servicios, y almacenando esta información en la base de dato: B.D de Grafos de info. de Usuarios.

El módulo de *categorización y visualización de la información de personalización* obtiene la información a partir de las dos bases de datos de grafos en donde se encuentran los grafos (árboles) de los requerimientos de usuarios y el árbol de las PNF. El submódulo del algoritmo de emparejamiento de grafos (Emparejador de Grafos) es el encargado del proceso de comparación de la información, el resultado de este proceso es un nuevo grafo (M) que es almacenado en la tercera base de datos.

El proceso de categorización (*ranking*) obtiene información del grafo M a través de una interfaz Web de la base de datos. Por último, el submódulo de visualización presenta al usuario el listado de los servicios que más se ajustan a sus requerimientos.

Cada módulo de la arquitectura tiene diferente tiempo de trabajo o ejecución, la información del usuario es obtenida y procesada en diferentes momentos, ya que el proceso de creación de la descripción del servicio (módulo verde) depende del número de servicios del repositorio y la variación de los parámetros de éstos. Así mismo, los demás módulos pueden ser invocados o ejecutados dependiendo de las peticiones del usuario y la variación de los requerimientos de éste.

3.2 Validación

Para probar la arquitectura se construyó un prototipo utilizando Java. Para los repositorios de grafos se utilizó una base de datos Neo4J versión 1.2, la cual es una base de datos completamente transaccional que almacena estructuras de datos como grafos. Para obtener información real de las PNF de los servicios Web, se recopilaron datos reales de *tiempo de respuesta y porcentaje de dis-*

```

1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:j.0="http://purl.org/vocab/relationship/"
4   <rdf:Description rdf:about="http://qos/confidentialityMetric">
5     <j.0:metricOf>
6       <rdf:Description rdf:about="http://qos/confidentiality">
7         <j.0:leafOf>
8           <rdf:Description rdf:about="http://qos/security">
9             <j.0:dimensionOf rdf:resource="http://qos/qos/">
10            </rdf:Description>
11          </j.0:leafOf>
12        </rdf:Description>
13      </j.0:metricOf>
14    </rdf:Description>
15    <rdf:Description rdf:about="http://qos/latencyValue">
16    <rdf:Description rdf:about="http://qos/bandwidthValue">
17    <rdf:Description rdf:about="http://qos/repairTimeValue">
18    <rdf:Description rdf:about="http://qos/latencyMetric">
19    <rdf:Description rdf:about="http://qos/ipValue">
20
21  101 </rdf:Description>
22  102 <rdf:Description rdf:about="http://qos/responseTimeValue">
23    <j.0:valueOf rdf:resource="http://qos/responseTime/">
24    <j.0:valueLiteralOf rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
25      >317.0</j.0:valueLiteralOf>
26    </rdf:Description>
27    <rdf:Description rdf:about="http://qos/service">
28      <j.0:nameOf rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
29        >Authentication</j.0:nameOf>
30      <j.0:descriptionOf rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
31        >Set child account's password. Change account's password. Logout of the given sess
32      <j.0:idOf rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

Figura 7. Estructura de un archivo RDF de la información de PNF de un servicio.

Fuente: elaboración propia

ponibilidad desde el portal *http://service-finder.eu*, utilizando el submódulo *Crawler Creador de PNF*; con este módulo fueron extraídos 122 servicios de diferentes categorías predefinidas por ese portal. Cada información es almacenada en un archivo RDF, cuya estructura sigue el trabajo propuesto en [12] y del cual se presenta un fragmento en la figura 7.

Con la información de las PNF se creó el grafo de servicios que contiene 122 servicios, lo cual genera un total de 733 nodos y 732 relaciones, teniendo en cuenta todos los nodos de todos los árboles. Para la visualización de este grafo se utilizó la herramienta Neoclipse. De igual manera, el grafo de los requerimientos de los usuarios fue creado utilizando información promedio de los

Tabla 2. Datos experimentales (parámetros requeridos por el usuario)

Número de servicios	122
Tiempo de respuesta del usuario: VALOR	250
Tiempo de respuesta del usuario: METRICA	Milliseconds
Porcentaje de disponibilidad: VALOR	97
Porcentaje de disponibilidad: METRICA	%
I	{1,1}, {2,2}, {5,5}, {7,7}, {10,10}, {20,20}, {50,50}, {70,70}, {100,100}

Fuente: elaboración propia

requerimientos de tiempo de respuesta y porcentaje de disponibilidad, obtenida igualmente desde: <http://demo.service-finder.eu/>.

A pesar de que la comparación para la creación del grafo M , es uno a uno entre los grafos de usuario y PNF, el rendimiento del algoritmo fue aceptable. Por ejemplo, en un computador portátil con procesador Intel Core 2 Duo @2200Mhz y 4 GB de memoria RAM, el proceso completo tomó

aproximadamente un promedio entre 40 a 50 segundos en todas las pruebas.

Fueron efectuadas diferentes medidas para obtener la métrica de *Recall*, la cual es definida como la fracción de documentos relevantes que son recuperados por un algoritmo y es utilizado como un factor para medir la calidad estadística en el análisis de sistemas de recuperación de información [24]; los datos base del experimento están consignados en la tabla 2. Para cada variación del parámetro I en $S = \{10, 24, 48, 87, 122\}$ se tomó el número de servicios relevantes con respecto al tiempo de respuesta y porcentaje de disponibilidad, y se calculó el parámetro de *Recall* utilizando dicha información.

3.3 Análisis de resultados

Los análisis de los parámetros de *Recall* fueron realizados teniendo en cuenta la figura 8, que representa los conjuntos de servicios recuperados los cuales tienen o no importancia para el usuario. En la figura 8, en la zona denominada A, se

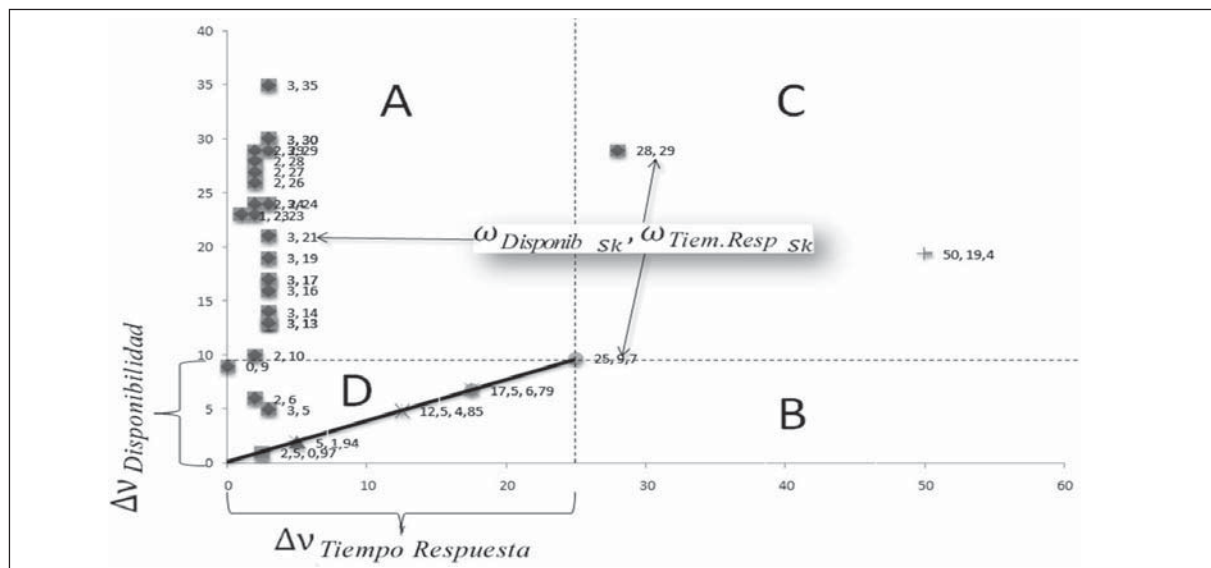


Figura 8. Zonas para determinar el parámetro de precisión.

Fuente: elaboración propia

encuentran los servicios que son relevantes para los requerimientos del parámetro de *tiempo de respuesta*, pero que al mismo tiempo no son relevantes para los requerimientos del usuario dado el parámetro de *disponibilidad*. En la zona B los servicios no son relevantes para los requerimientos de *tiempo de respuesta* pero lo son para los requerimientos de *disponibilidad*. En la zona C se encuentran parámetros que están fuera de los límites de los valores de *tolerancia* de los requerimientos del usuario; y finalmente, en la zona D se encuentran los servicios que cumplen con las dos condiciones de los requerimientos del usuario. Cada servicio está ubicado dentro un espacio geométrico cuyas coordenadas son dadas por el par $(\omega_{avilab_{sk}} \ \omega_{respTime_{sk}})$

Para calcular el parámetro de *Recall* se utilizó la expresión siguiente:

$$Recall = \frac{VP}{VP+FN} = \frac{D}{A+B+D} \quad (8)$$

En la cual VP representan los Verdaderos Positivos, que son los resultados que satisfacen todos los requerimientos de los usuarios y en estadística muestran los verdaderos y útiles resultados.

Los FN representan los datos Falsos Negativos que son aquellos datos que no concuerdan con los requerimientos del usuario pero que son recuperados por el sistema.

Las medidas obtenidas del parámetro *Recall* de la arquitectura, utilizando la información de prueba y las métricas se resumen en la figura 9.

4. CONCLUSIONES

El análisis de los trabajos relacionados que resumen el estado del arte, al momento de la investigación, mostró diferentes tipos de brechas en la descripción de los requerimientos no funcionales de los usuarios, las existentes en la descripción de las propiedades no funcionales de los servicios y en el proceso de emparejamiento de servicios basado en propiedades no funcionales. En este trabajo se mejoró el método de descripción de servicios Web presentado en [5], [6], y [11], utilizando un metamodelo genérico, flexible y extendible. Así mismo, se desarrolló un método para estructurar los requerimientos de un usuario, mejorando el trabajo de [16] y utilizando una aproximación

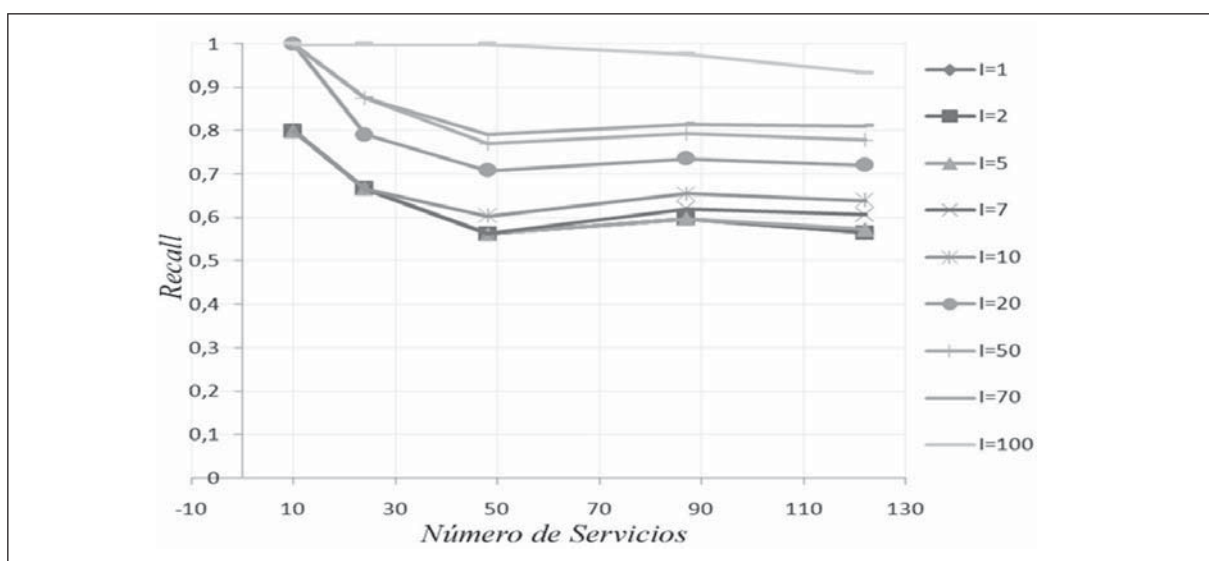


Figura 9. Gráfico de Recall variando el parámetro l.

Fuente: elaboración propia

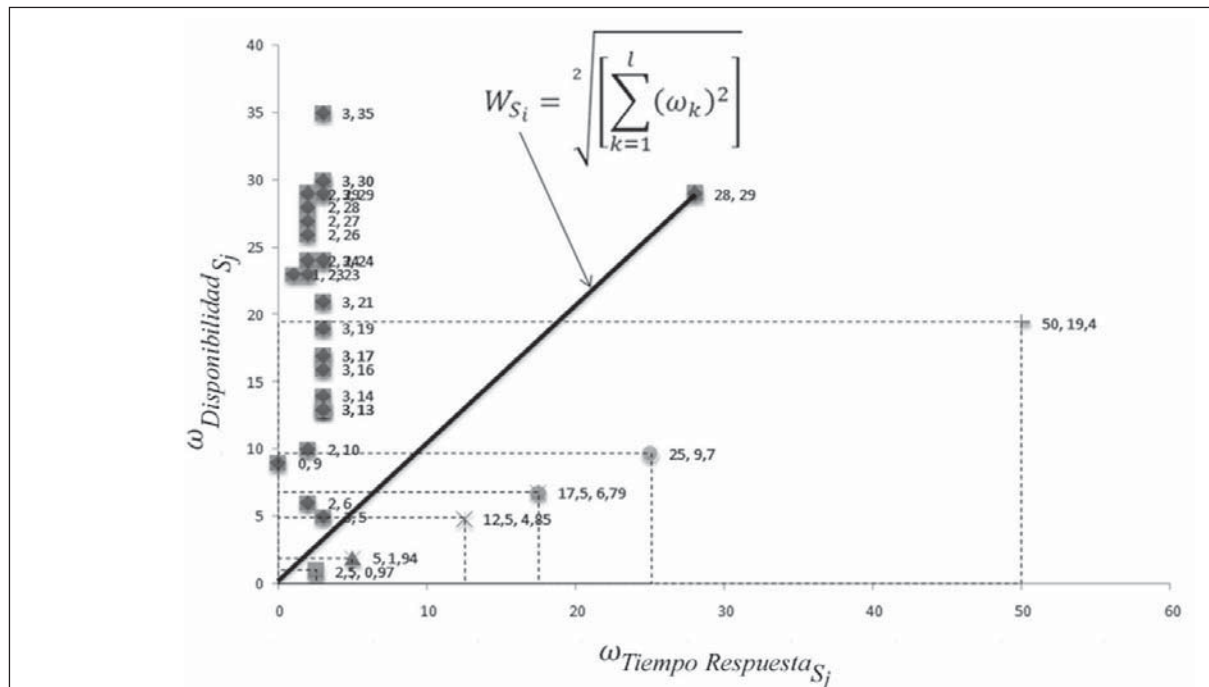


Figura 10. Análisis de la distancia Euclidiana de un servicio Web y las áreas de los valores de tolerancia.

Fuente: elaboración propia

flexible por medio del vector de importancia y el método de distancia relativa propuesta aquí.

En cuanto a las conclusiones basadas en los resultados experimentales, analizando el gráfico de los resultados del parámetro *Recall*, se puede establecer que dicha medida claramente varía con el parámetro *I*.

Cada medida es tomada a partir de la información de *tiempo de respuesta* y *porcentaje de disponibilidad* y los valores de la tabla de datos experimentales (tabla 2). Con esta información se crea un área que contiene todos los servicios que tienen peso relevante (zona D), y utilizando la distancia euclidiana se puede hacer un rango de todos los servicios relevantes.

El anterior análisis demuestra que, a medida que el vector de importancia es modificado, la recuperación de servicios es más precisa, esto es, con

valores cercanos a $I=1$ el valor del número de Verdaderos Positivos es menor. La experimentación permitió analizar un comportamiento teórico de los Verdaderos Positivos y de los Falsos Negativos, pero requiere, para un proyecto futuro, tener una comparación de resultados cualitativa con grupos de prueba para poder establecer una medida denominada Precisión la cual es utilizada al igual que Recall en el análisis de sistemas de recuperación. Se puede concluir que, en este proyecto, se utilizó una aproximación matemática para resolver un problema tecnológico, utilizando grafos (árboles) para modelar propiedades no funcionales de los servicios, así como también para modelar información de los requerimientos de los usuarios.

El proceso de emparejamiento entre los grafos fue efectuado utilizando una variación del algoritmo de emparejamiento VF2, el cual fue modificado para que trabaje con árboles siendo, por

lo tanto, más eficiente, dado que se eliminan los métodos de verificación de nodos y vértices incidentes.

En este trabajo se definió la *distancia* de la información de requerimientos y PNF (hojas de los árboles) utilizando el parámetro PESO dentro del grafo de emparejamiento. Se categorizaron los servicios más relevantes utilizando una aproximación geométrica basada en el uso de la teoría de espacios euclidianos, manipulando los l parámetros de los servicios como coordenadas en un espacio de l dimensiones.

Se concluyó que la variación del parámetro l es útil para el mejoramiento del comportamiento del sistema y de las medidas del parámetro *Recall*.

En contextos de servicios Web restringidos a unas pocas categorías de parámetros, puede ser posible definir los valores del vector l de manera precisa, por ejemplo en el caso del contexto de las telecomunicaciones y más exactamente en el caso de servicios de video digital sobre redes IP, la ITU-T se han definido ciertas medidas de

QoS que, al igual que en otros contextos, pueden reducir el número de servicios no relevantes recuperados.

5. FINANCIAMIENTO

Este proyecto fue parcialmente financiado por el Departamento de Telemática de la Universidad del Cauca; los recursos de movilidad del Ingeniero Guerrero en la Universidad del Malmö – Suecia fueron a través del Programa COOPEN Erasmus Mundus.

6. AGRADECIMIENTOS

Los autores expresan su agradecimiento a la Universidad del Cauca por su apoyo Institucional y al Programa Erasmus Mundus por financiar los estudios del Ingeniero Guerrero en la Universidad de Malmö - Suecia. Este proyecto fue avalado por el Departamento de Telemática de la Universidad del Cauca y el Comité de Investigaciones de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la misma Universidad.

REFERENCIAS

- [1] A. Tsalgatidou and T. Pilioura, “An overview of standards and related technology in web services”, *Distributed and Parallel Databases*, Vol. 12, No. 2, 2002, pp. 135–162.
- [2] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Research Directions in Service-Oriented Computing”, in *4a IC-SOC 2006*, Vienna Austria, 2006, p. 28.
- [3] J. J. O’Sullivan, *Towards a precise understanding of service properties*, [Tesis], Queensland University of Technology, Brisbane: Queensland Australia, 2006.
- [4] P. Li, M. Comerio, A. Maurino, and F. D. Paoli, “An Approach to Non-functional Property Evaluation of Web Services”, in *IEEE International Conference on Web Services, Washington, DC, USA, 2009*, pp. 1004-1005.
- [5] D. Martin et al., “OWL-S: Semantic Markup for Web Services”, *World Wide Web Consortium (W3C) member Submis-*

- sion, 2004. [Online]. Available: <http://www.w3.org/Submission/OWL-S>.
- [6] D. Martin, M. Burstein, O. Lassila, and D. McDermott, "OWL-S : Semantic Markup for Web Services v1.2", *The DARPA Agent Markup Language Homepage*, 2009. [Online]. Available: <http://www.ai.sri.com/daml/services/owl-s/1.2/overview>.
- [7] J. O'Sullivan, D. Edmond, and A. H. M. ter Hofstede, "Formal description of non-functional service properties, reporte de: Related initiatives of Web Service Modeling Ontology (WSMO)". Brisbane: Australia, 2005. [Online]. Available: <http://www.wsmo.org/papers/OSullivanTR2005.pdf>
- [8] S. Aier, P. Offermann, M. Schönherr, and C. Schröpfer, "Implementing non-functional service descriptions in SOAs", in *Trends in Enterprise Application Architecture*, Springer Berlin / Heidelberg, 2007, pp. 40–53.
- [9] I. Toma, D. Foxvog, F. D. Paoli, M. Comerio, M. Palmonari, and A. Maurino, "Non-functional properties in Web services, reporte del WSMO Final Draft", *DERI, D28.4v0.1*, 2008. [Online]. Available: <http://www.wsmo.org/TR/d28/d28.4/v0.1>
- [10] J. Domingue, D. Fensel, H. Lausen, and A. Polleres, "Web Service Modeling Ontology Primer", *World Wide Web Consortium (W3C) member Submission*, 2005, [Online]. Available: <http://www.w3.org/Submission/WSMO-primer>
- [11] J. O. Sullivan, "Towards a Precise Understanding of Service Properties," in *ICEIS Doctoral Consortium*, Queensland Australia, 2006.
- [12] E. Guerrero, J. C. Corrales, and R. Raggia, "Recuperación de Servicios basada en la Personalización del proceso de Descubrimiento," en *EATIS-2010*, Panamá, 2010.
- [13] P. Wang, "QoS-aware web services selection with intuitionistic fuzzy set under consumer 's vague perception", *Expert Systems With Applications*, Vol. 36, No. 3, pp. 4460-4466, 2009.
- [14] W. Abramowicz, K. Haniewicz, M. Kaczmarek, R. Palma, and D. Zyskowsxi, "NFP ontology for discovery and sharing Web services in distributed registries", en *AINAW-2008*, Okinawa, Japan, 2008.
- [15] B. Pernici and S. Hossein, Siadat, "Adaptation of Web Services Based on QoS Satisfaction" in *ICSOC 2010 International Workshops*, San Francisco, California, 2010.
- [16] L.-H. Vu, M. Hauswirth, and K. Aberer, "Towards p2p-based semantic web service discovery with qos support," in *Third International Conference on Business Process Management*, Nancy, France, 2006.
- [17] O. Choi, H. Choi, Z. Lee, and S. Han, "Semantic Web Services Discovery System with QoS for Enhanced Web Services Quality," in *CDVE 2007*, Shanghai, China, 2007.
- [18] C. E. Serrano, *Modelo Integral para el Profesional en Ingeniería*. Popayán, Colombia: Ediciones Universidad del Cauca, 2005.
- [19] R. Diestel, *Graph Theory*. Hamburg. Germany: Springer-Verlag Heidelberg, 2005.

- [20] A. Ferro et al., “NetMatch : a Cytoscape plugin for searching biological networks”, *Bioinformatics*, Vol. 23, No. 7, pp. 910-912.
- [21] D. W. Williams, J. Huan, and W. Wang, “Graph Database Indexing Using Structured Graph Decomposition” in *23rd International Conference on Data Engineering*, Istanbul, Turkey, 2007.
- [22] K. V. Mital and C. Mohan, *Optimization methods in operations research and system analysis*. New Delhi, India: New Age International (P) Ltd., 2007.
- [23] Y. J. Huang, R. Powers, and G. T. Montelione, “Protein NMR recall, precision, and F-measure scores (RPF scores): structure quality assessment measures based on information retrieval statistics”, *Journal of the American Chemical Society*, Vol. 127, No. 6, 2005, pp. 1665–1674.
- [24] *Quality of service ranking and measurement methods for digital video services delivered over broadband IP networks*, ITU T Recommendation J.241 - International Telecommunication Union ITU-T, 2005.